Rich Web Application Technologies

Lab Sheet 2

C14347391 | Ben Shanahan

*Question 1:*

*HTML :*

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width">  
 <title>Calculator</title>  
 <link rel="stylesheet" type="text/css" href="calCSS.css">  
 <script src="calc.js">  
  
 </script>  
</head>  
<body>  
<div id="calculator">  
 <form name="calculator">  
 <input id="input" name="input" type="text" readonly="readonly" placeholder="0">  
 </form>  
 <div class="flex-container">  
 <button class="flex-item" id="div" onclick="screenAdd('/')">÷</button>  
 <button class="flex-item" id="7" onclick="screenAdd(7)">7</button>  
 <button class="flex-item" id="8" onclick="screenAdd(8)">8</button>  
 <button class="flex-item" id="9" onclick="screenAdd(9)">9</button>  
 <button class="flex-item" id="multi" onclick="screenAdd('\*')">x</button>  
 <button class="flex-item" id="4" onclick="screenAdd(4)">4</button>  
 <button class="flex-item" id="5" onclick="screenAdd(5)">5</button>  
 <button class="flex-item" id="6" onclick="screenAdd(6)">6</button>  
 <button class="flex-item" id="min" onclick="screenAdd('-')">-</button>  
 <button class="flex-item" id="1" onclick="screenAdd(1)">1</button>  
 <button class="flex-item" id="2" onclick="screenAdd(2)">2</button>  
 <button class="flex-item" id="3" onclick="screenAdd(3)">3</button>  
 <button class="flex-item" id="plus" onclick="screenAdd('+')">+</button>  
 <button class="flex-item" id="0" onclick="screenAdd(0)">0</button>  
 <button class="flex-item" id="dot" onclick="screenAdd('.')">.</button>  
  
 <button class="flex-item" id="clear" onclick="clear()">C</button>  
  
 <button class="flex-item" id="equal" onclick="getAnswer()">=</button>  
  
 </div>  
</div>  
</body>  
</html>

*CSS :*

input[type=text] {  
 font-size: 2em;  
 text-align: right;  
 border-radius: 5px;  
 width: 240px;  
 height: 35px;  
 margin-bottom: 10px;  
 padding: 1%;  
}  
button {  
 font-size: 1.1em;  
 border-radius: 5px;  
 width: 50px;  
}  
.flex-container {  
 display: inline-flex;  
 flex-flow: row wrap;  
 justify-content: space-between;  
}  
.flex-item {  
 text-align: center;  
 padding: .5em;;  
 font-size: 1em;  
 flex-grow: 1;  
 margin: 0 5px 5px 0;  
}  
#calculator {  
 width: 250px;  
 font-size: 100%;  
 border: 1px solid black;  
 padding: 10px;  
 border-radius: 7px;  
 margin-left: auto;  
 margin-right: auto;  
 font-family: Verdana;  
}

*JavaScript :*

**function** screenAdd (num) {  
 document.calculator.input.value = document.calculator.input.value + num  
}  
**function** getAnswer() {  
 document.calculator.display.value=eval(document.calculator.input.value)  
}  
**function** clear() {  
 document.calculator.display.value = 0  
}

*Question 2:*

*HTML :*

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width">  
 <title>RW\_JSON</title>  
</head>  
<body>  
<script src="RW\_JSON.js" type="text/javascript"></script>  
<p id = "username"></p>  
<br>  
<hr>  
<p id = "geos"></p>  
<br>  
<hr>  
<p id = "users"></p>  
<br>  
<hr>  
<p id = "address"></p>  
</body>  
</html>

*JavaScript :*

root = "http://jsonplaceholder.typicode.com/";  
userpath = "users";  
  
// XMLHttpRequest class  
**const** xhr = **new** XMLHttpRequest();  
xhr.onreadystatechange = () => {  
 **if** (xhr.readyState === XMLHttpRequest.DONE) {  
 **if** (xhr.status === 200) {  
 **let** response = JSON.parse(xhr.responseText);  
 console.log(response);  
  
  
 // 1-- User name Array  
 **let** usernameArray = Object.keys(response).map(**function**(key) {  
 **return** response[key].username;  
 });  
 console.log(usernameArray);  
 usernameArray.toString();  
 document.getElementById("username").innerHTML = usernameArray;  
  
  
 // 2-- Geo objects Array  
 **let** geoArray = Object.keys(response).map(**function**(key) {  
 **let** lat = {lat: response[key].address.geo.lat + ', ' + response[key].address.geo.lng};  
 **return** lat;  
 });  
 console.log(geoArray);  
 **let** geoArray2 = geoArray.map(Number);  
 document.getElementById("geos").innerHTML = geoArray2;  
  
  
  
 // 3-- User details Array  
 **let** userArray = Object.keys(response).map(**function**(key) {  
 **let** user = [{name: response[key].name},  
 {id: response[key].id},  
 {companyname: response[key].company.name},  
 {zip: response[key].address.zipcode}];  
 **return** user;  
 });  
 console.log(userArray);  
  
  
 // 4-- Location Array  
 **let** addressZip = Object.keys(response).filter(**function**(key) {  
 **return** response[key].address.zipcode.charAt(0) === '5';  
 }).map(**function**(key) {  
 **return** response[key].address;  
 });  
 console.log(addressZip);  
  
 // 5-- Product Array  
 **let** total = Object.keys(response).reduce(**function**(total, num, key) {  
 **return** total \* response[key].id;  
 }, 1);  
 console.log(total);  
 }  
 }  
};  
  
xhr.open("GET", root + userpath);  
xhr.send();

*Question 3:*

*HTML:*

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width">  
 <title>User Profile</title>  
 <link rel="stylesheet" type="text/css" href="gitZ.css">  
</head>  
<body>  
<header>  
 <input type="text" placeholder="Username">  
 <button id="submit">Search</button>  
</header>  
  
<div class="flexboxDiv">  
 <section>  
 <h2>User Profile</h2>  
 <div id="imageDiv">  
 <img id="profile-image" alt="" src="">  
 </div>  
 <ul id="Prof">  
 <li><div id="Lname">Name</div></li>  
 <li><div id="Luname">Username</div></li>  
 <li><div id="Lemail">Email</div></li>  
 <li><div id="Lloc">Location</div></li>  
 <li><div id="Lgist">Number of Gists</div></li>  
 </ul>  
 </section>  
  
 <section>  
 <h2>User Repos</h2>  
 <div id="repoDiv">  
 <ul id="repoList"></ul>  
 </div>  
 </section>  
</div>  
  
<script src="gitZ.js" type="text/javascript"></script>  
</body>  
</html>

*CSS:*

body {  
 width: 800px;  
}  
header {  
 margin: 10px;  
}  
ul {  
 list-style: none;  
 align-content: center;  
}  
ul li {  
 border: 1px solid #000000;  
 height: 50px;  
 padding: 2px;  
 text-align: left;  
}  
ul li div {  
 padding: 8px;  
}  
img {  
 width: 250px;  
 height: 250px;  
 display: block;  
 margin: 0 auto;  
}  
input[type=text] {  
 outline: none;  
 border: 1px solid #000000;  
 width: 70%;  
 height: 20px;  
 margin-right: 40px;  
 padding: 5px;  
}  
#submit {  
 width: 150px;  
 height: 30px;  
 float: right;  
 border-radius: 10px;  
}  
  
  
#imageDiv {  
 border: 1px solid #000000;  
}  
#Prof li div {  
 padding-top: 15px;  
}  
#repoList li {  
 height: 30%;  
  
}  
  
.flexboxDiv {  
 display: flex;  
 height: auto;  
 flex-flow: row;  
 justify-content: center;  
 overflow: scroll;  
 width: 100%;  
 margin-right: 20px;  
 border: 1px solid #000000;  
}

*JavaScript :*

**const** URL = "https://api.github.com/users/";  
  
**var** search = document.querySelector("button[id=submit]"); //returns first element that matches submit id  
**var** repos = document.getElementById("repoList"); //gets repo list  
  
**var** options = {  
 method : 'GET',  
 headers: [ {name: "Content-Type", value: "application/json"} ]  
 };  
  
**function** fetch(url, options) {  
 **return new** Promise(**function**(resolve, reject) {  
  
 **var** xRequest = **new** XMLHttpRequest();  
 xRequest.open(options.method, url); //oppens http connection  
  
 xRequest.onload = **function**() {  
 **if** (xRequest.status == 200) { //200 = successful request  
 resolve(xRequest.response);  
 } **else** {  
 reject(Error(xRequest.statusText));  
 }  
 };  
 xRequest.onreadystatechange = **function**(e) {  
 **if** (**this**.readyState === 4 && **this**.status === 200) //response is sent & request was successful  
 {  
 **try** {  
 }  
 **catch** (error) {  
 **return false**;  
 }  
 }  
 };  
 xRequest.send();// Send request  
 });  
}  
  
search.onclick = **function**(s) {  
 **var** input = document.querySelector("input[type=text]").value;  
 **if** (input != **null**) {  
 **while** (repos.firstChild) {  
 repos.removeChild(repos.firstChild);  
 }  
  
 fetch(URL + input, options).then(**function**(data) { //fetches from url different elements ans saves them to designated vars  
 **var** json = JSON.parse(data);  
 setProfileImage(json.avatar\_url);  
 setProfileName(json.name);  
 setProfileUsername(json.login);  
 setProfileEmail(json.email);  
 setProfileLocation(json.location);  
 setProfileGists(json.public\_gists);  
 getRepos(json.repos\_url);  
 });  
 }  
};  
  
**function** getRepos(repoURL) {  
 fetch(repoURL, options).then(**function**(data) {  
 **var** json = JSON.parse(data);  
 **for** (**var** i = 0; i < 100; i++) { //loads up to 100 repos  
 addRepoRow(json[i].name, json[i].description)  
 }  
 }, **function**(err) {  
 //console.log(err);  
 console.log("The request has failed!");  
 });  
}  
  
**function** addRepoRow(name, description) { //add repo layout  
 **var** newRepoItem = document.createElement("li");  
 **var** RepoName = document.createElement("div");  
 **var** RepoDesc = document.createElement("div");  
 repos.appendChild(newRepoItem);  
 newRepoItem.appendChild(RepoName);  
 newRepoItem.appendChild(RepoDesc);  
 RepoName.innerHTML = "Name: " + name;  
 RepoDesc.innerHTML = "Description: " + description;  
}  
  
**function** setProfileImage(url) { //fetch user info & user info layout  
 document.getElementById("profile-image").src = url;  
}  
**function** setProfileName(name) {  
 document.getElementById("Lname").innerHTML = "Name: " + name;  
}  
**function** setProfileUsername(username) {  
 document.getElementById("Luname").innerHTML = "Username: " + username;  
}  
**function** setProfileEmail(email) {  
 document.getElementById("Lemail").innerHTML = "Email: " + email;  
}  
**function** setProfileLocation(location) {  
 document.getElementById("Lloc").innerHTML = "Location: " + location;  
}  
**function** setProfileGists(public\_gists) {  
 document.getElementById("Lgist").innerHTML = "Number of Gists: " + public\_gists;  
}

*Lecture Review Questions:*

What does it mean for a data structure to be described as a functor? Give a code example in Javascript in your explanation.

A functor is a data structure that can be mapped over.

*console.log([ 1, 2, 3 ].map(x => x \* 2))*

= [1, 4, 6]

This is an array being mapped over by the function x\*2

Describe how the flexbox model works in CSS

A flexbox ensures that elements on a page behave predictably when the page is altered due to screen size or display devices. A flexbox does not use floats nor does its margin’s collapse with the margins of its contents and so is an improvement to a block model.